



# **DEVELOPMENT OF THE ARCHITECTURAL SIMULATION MODEL FOR FUTURE LAUNCH SYSTEMS AND ITS APPLICATION TO AN EXISTING LAUNCH FLEET**

## **Final Report**

**Phase I:** October 2003 – July 2004

**Phase II:** August 2004 – February 2005

**ODU Project No.: 140453**

May 3, 2005

Submitted by:

Ghaith Rabadi, Ph.D.

Department of Engineering Management & Systems Engineering  
Old Dominion University  
241 Kaufman Hall  
Norfolk, VA 23529  
[grabadi@odu.edu](mailto:grabadi@odu.edu)

Submitted to:

Trina Chytka and Richard Brown

Vehicle Analysis Branch  
Aerospace Systems, Concepts and Analysis Competency  
National Aeronautics and Space Administration  
Langley Research Center, Mail Stop 365  
Hampton, Virginia 23681

## Table of Contents

Table of Contents .....	2
List of Figures .....	3
List of Tables .....	4
<b>PHASE I:</b> .....	5
<b>PHASE I:</b> .....	5
<b>DEVELOPMENT OF A DISCRETE-EVENT SIMULATION MODEL TO ASSESS OPERATIONS AND SUPPORT COST OF LAUNCH SYSTEM ARCHITECTURES</b> .....	5
Abstract .....	5
Introduction .....	6
Problem Statement .....	6
Methodology .....	6
The Architectural Model Framework .....	7
The Architectural Model .....	11
Processing Times .....	15
Cost Information .....	15
<b>PHASE II:</b> .....	18
<b>RESEARCH EXTENSION OF THE ARCHITECTURAL SIMULATION MODEL FOR EXISTING AND FUTURE LAUNCH SYSTEMS</b> .....	18
Abstract .....	18
Introduction .....	19
Architectural Model Extension and Modification .....	19
Modularity Study .....	21
Conclusions .....	23
Future Work .....	23

## List of Figures

Figure 1. Launch Systems Structure .....	9
Figure 2. An Example of An Architecture Driven by Manifest Requirements .....	10
Figure 3. Process Flow Chart.....	11
Figure 4. Process Flow in the Model .....	12

## List of Tables

Table 1. Launch Elements.....	13
Table 2. Manifest Sample .....	14
Table 3. Destination Code.....	14
Table 4. Lunar Mission Component Descriptions .....	22
Table 5. Modularity Level Element Breakdowns.....	22

## **PHASE I: DEVELOPMENT OF A DISCRETE-EVENT SIMULATION MODEL TO ASSESS OPERATIONS AND SUPPORT COST OF LAUNCH SYSTEM ARCHITECTURES**

### **Abstract**

A significant portion of lifecycle costs for launch vehicles are generated during the operations phase. Research indicates that operations costs can account for a large percentage of the total life-cycle costs of reusable space transportation systems. These costs are largely determined by decisions made early during conceptual design. Therefore, operational considerations are an important part of vehicle design and concept analysis process that needs to be modeled and studied early in the design phase. However, this is a difficult and challenging task due to uncertainties of operations definitions, the dynamic and combinatorial nature of the processes, and lack of analytical models and the scarcity of historical data during the conceptual design phase.

Ultimately, NASA would like to know the best mix of launch vehicle concepts that would meet the missions' launch dates at the minimum cost. To answer this question, we first need to develop a model to estimate the total cost, including the operational cost, to accomplish this set of missions. In this project, we have developed and implemented a discrete-event simulation model using ARENA (a simulation modeling environment) to determine this cost assessment. Discrete-event simulation is widely used in modeling complex systems, including transportation systems, due to its flexibility, and ability to capture the dynamics of the system.

The simulation model accepts manifest inputs including the set of missions that need to be accomplished over a period of time, the clients (e.g., NASA or DoD) who wish to transport the payload to space, the payload weights, and their destinations (e.g., International Space Station, LEO, or GEO). A user of the simulation model can define an architecture of reusable or expendable launch vehicles to achieve these missions. Launch vehicles may belong to different families where each family may have its own set of resources, processing times, and cost factors. The goal is to capture the required resource levels of the major launch elements and their required facilities. The model's output can show whether or not a certain architecture of vehicles can meet the launch dates, and if not, how much the delay cost would be. It will also produce aggregate figures of missions cost based on element procurement cost, processing cost, cargo integration cost, delay cost, and mission support cost. One of the most useful features of this model is that it is stochastic where it accepts statistical distributions to represent the processing times mimicking the stochastic nature of real systems.

## **Introduction**

A significant portion of lifecycle costs for launch vehicles are generated during the operations phase. Research indicates that operations costs can account for a large percentage of the total life-cycle costs for reusable space transportation systems. These costs are largely determined by decisions made early during conceptual design. Therefore, operational considerations are an important part of vehicle design and concept analysis process that needs to be modeled and studied early in the design phase. However, this is a difficult and challenging task due to uncertainties of operations definitions, the dynamic nature of the processes, and lack of analytical models and the scarcity of historical data during the conceptual design phase. In this project, we develop and implement a simulation model to assess the cost of future vehicle designs, including operational costs, that would meet the manifest schedule.

## **Problem Statement**

The problem addressed in this project is that a set of missions to transport cargo (or payloads) to space needs to be carried out. The cargo may have different weights and destinations. There might be different types of space launch vehicles to accomplish these missions but at different costs. Ultimately, NASA would like to know the best mix of launch vehicle concepts that would meet the missions' launch dates at the minimum cost. To answer this question, however, we first need to have a way to estimate the cost of accomplishing the missions when it is known beforehand which type of vehicles are going to fly which missions, as well as the level of resources required to support these vehicles. Once this is done, the next phase would be to develop an optimization model that uses this cost model to find the best assignment of vehicles to missions and the best resource levels that minimize the total cost over a defined life cycle and meet the missions' launch dates.

In this project, we addressed the first phase only. That is, we developed and implemented a model to estimate the cost of accomplishing the manifest given that we know which vehicles are going to fly which missions and what level of resources available to process these vehicles.

## **Methodology**

In this research, a discrete-event simulation (DES) model called the *Architectural Model* was developed to address this problem. DES is widely used to study complex systems including space transportation systems because of its flexibility, and ability to capture the dynamics of the system. It is also considered a stochastic modeling technique where the inputs to the model can take the form of statistical distributions, and events can be of probabilistic nature.

DES is a numerical computer-based simulation technique that has been under development for the past few decades. General computer-based simulation has roots that trace back to the 1950s when computer programming became popular, but not until the past couple of decades has DES become a viable technique. DES' most vital benefit is that it combines the relative ease and flexibility of computer programming with the crucial results of statistical analysis. The late 1970s and early 1980s saw the emergence

of numerous DES simulation languages such as GPSS, SIMSCRIPT, SLAM, and SIMAN that began to capture the effects of combining statistics with computer programming. These languages became industry standards due to their applicability to almost any sort of engineering, manufacturing, or any other queue-based field. DES quickly became associated with the field of Industrial Engineering due to its inherent statistics foundation, as well as its popular application to manufacturing, transportation, and other logistics-based activities. The early simulation languages such as SLAM, SLAM II, and SIMAN were powerful, but required a heavy amount of programming and a significant learning curve. Due to the boom in computing power of the early 1990s along with an increase in graphical user interfaces (GUIs) popularity, advanced simulation products that combined the power of the early simulation languages such as SIMAN with the ease of use and reduced complexity of a graphical environment began to hit the market. In addition to having a graphical coding interface that reduced the learning curve required, some packages began to include sophisticated animation capabilities that not only assisted users in troubleshooting, but also gave users a way of demonstrating model logic and dynamics to others.

Rockwell Software's Arena is a package that has consistently been a popular product of the DES software industry for the past several years. Its popularity can be traced to its ability to provide useful results without requiring too significant of a learning curve. One of Arena's most beneficial traits is that users across the whole spectrum of skill-levels can use the product to generate useful results. This robustness is achieved by expanding upon an evolved version of the SIMAN language, meaning that Arena has been built upon the shoulders of an already successful product. Arena allows users to choose from various *modules* that are presented in various *templates* ranging from basic logic pieces to complex items such as conveyers and transporters. Each module represents a combination of SIMAN code that has been pre-packaged to allow the user to drag and drop pieces of code into the model without having to work with the code itself. In fact, an entry-level user can design, develop, and execute somewhat complex Arena models without having to type a single line of code. Arena also provides automatically generated reports at the end of simulation runs that can be modified in various ways in order to present specified resulting statistics.

The Architectural Model, which was implemented in Arena, is general enough to be applied to several launch system architectures to estimate the cost, and operation and support (O&S) requirements of new launch system architectures.

### **The Architectural Model Framework**

Before implementing the simulation model, it is necessary to design a framework within which we determine the modeling level of detail and model capabilities. This framework includes the following components:

- 1) Launch Systems Structure: The following structure definition is used in the development of the model:

*Element:* The basic building block for a Launch Vehicle Configuration (i.e., Booster, Orbiter, OMV, Cargo Carrier, etc.).

*Configuration:* A unique combination of elements from a Launch Family to form a vehicle capable of performing a specific mission.

*Family:* A Launch Vehicle and all of its variants, where each variant is a configuration.

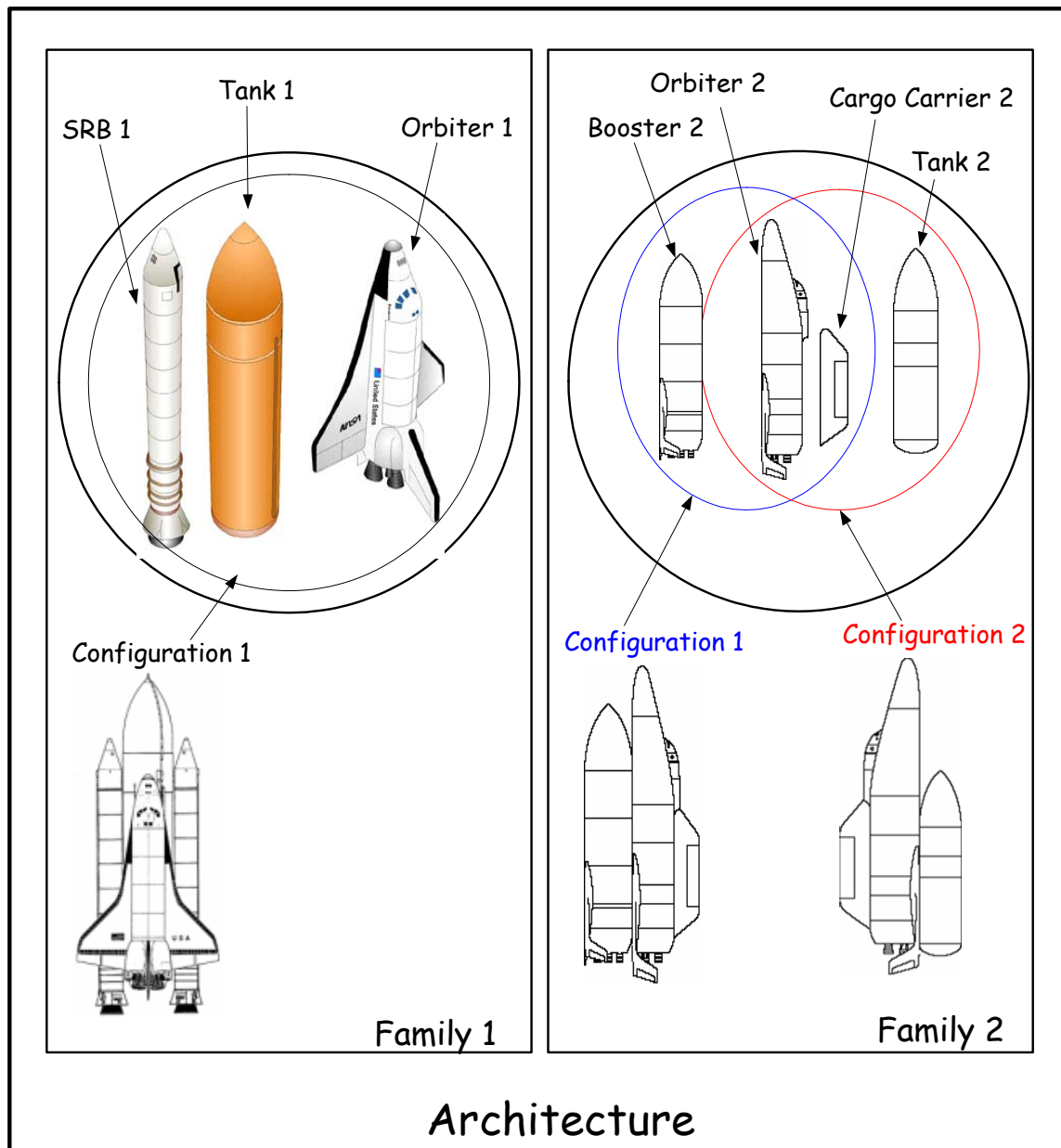
*Architecture:* an aggregation of individual Launch Families that represent the total launch capability available to support a manifest.

Figure 1 further illustrates the meaning of these terms using an example of an Architecture that encompasses two families: Family 1 and Family 2. In fact Family 1 is the Shuttle Family, which includes three element types: Orbiter 1, Tank 1 and SRB1 (Solid Rocket Booster). The integration of number of these elements results in a Configuration. In this example, the integration of Orbiter 1, Tank 1 and two of SRB1 results in Configuration 1, which is actually the Space Shuttle Vehicle. Family 2, on the other hand, is a future launch vehicle family that includes a different type of elements: Orbiter 2, Booster 2, Tank 2 and Cargo Carrier 2. Note here that the index of the elements (i.e., 2) is the family number. If Orbiter 2, Cargo Carrier 2, and Booster 2 are integrated, Configuration 2 will be generated. If, however, Orbiter 2, Cargo Carrier 2, and Tank 2 are integrated, Configuration 3 will be generated.

Within this framework, it is important to note the following design requirement:

- ❑ Elements can be either expendable or reusable. A Configuration (i.e., a vehicle) may consist of expendable Elements, reusable Elements, or a mixture of both.
- ❑ There could be unlimited number of Element types within a Family
- ❑ There could be unlimited number of Elements in a Configuration
- ❑ There could unlimited number of Configuration in an Architecture
- ❑ There could be unlimited number of Families within an Architecture
- ❑ There is a predefined set of generic Element types for all Architectures





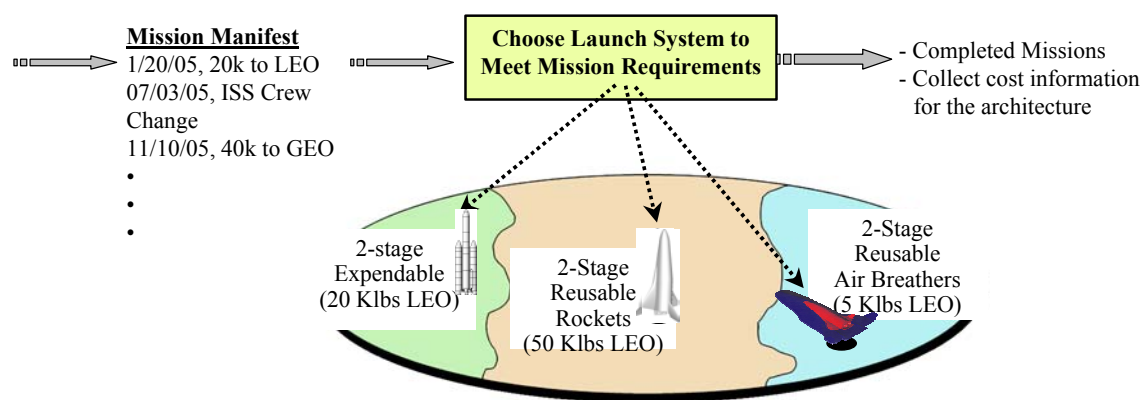
**Figure 1.** Launch Systems Structure

## 2) Manifest-driven Process

The objective of the model is to be able to compare the ability of alternate space transport architectures to meet a given manifest. Specific attributes targeted are cost and schedule. The model is driven by manifest requirements; given a manifest, which defines individual mission requirements, the model will determine the likelihood that the manifest can be met and at what cost. Mission requirements include payload to orbit weight, destination (GEO, LEO, ISS, Planetary, and Polar), launch date, customer, manned or unmanned mission, and Family and Configuration selection per mission.

Figure 2 shows an example of an architecture that may have 3 Configurations (or Vehicles) that belong to three different Families: a two-stage expendable vehicle capable of 20,000 pounds to LEO or 10,000 pounds to GEO; a two-stage man rated reusable vehicle capable of placing 50,000 pounds in LEO or 20,000 pounds to GEO; and a two-stage man rated reusable Air Breather capable of placing a 5,000 pound payload to LEO.

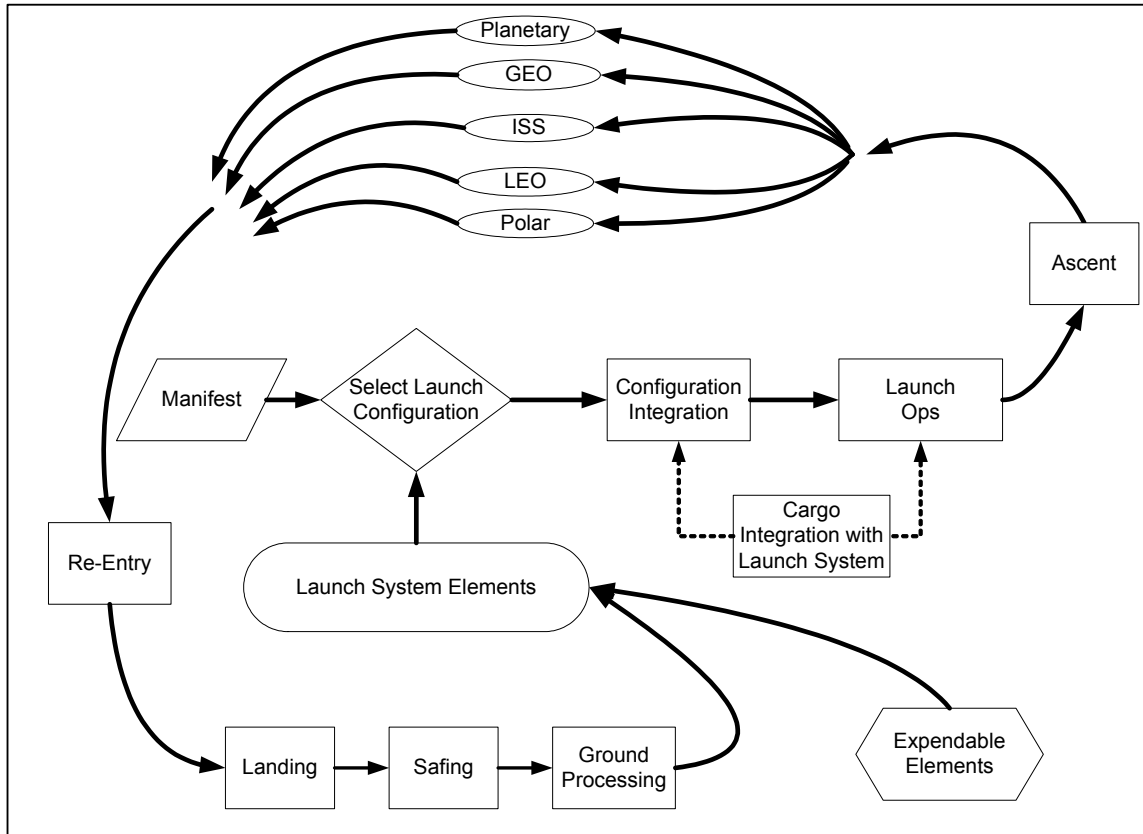
The manifest will include missions that need to deliver payloads with certain weight to certain destination at certain times. Based on vehicle selection, the model should produce information about the likelihood of meeting the manifest launch dates as well as the cost of accomplishing these missions.



**Figure 2.** An Example of An Architecture Driven by Manifest Requirements

### 3) Process Flow

Following the process flow illustrated in Figure 3, the model would use the manifest to make demands on systems (vehicles and their support resources) to be ready to launch at a predetermined date specified on the manifest. Once the elements are available and ready for integration, they are moved to the Configuration Integration to assemble the elements together and then move the vehicle to Launch Operations. Cargo Integration takes place either in the Configuration Integration or during Launch Operations. On launch date, the vehicle launches to go to Ascent then to Orbit as demanded by the manifest. Depending whether the vehicle includes expendable elements or not, these elements will be disposed and only reusable elements will land and be safed. Then the elements go through ground processing to be available for new missions to use. As they are released, new missions seize the necessary reusable elements and/or acquire new expendable elements if needed.



**Figure 3. Process Flow Chart**

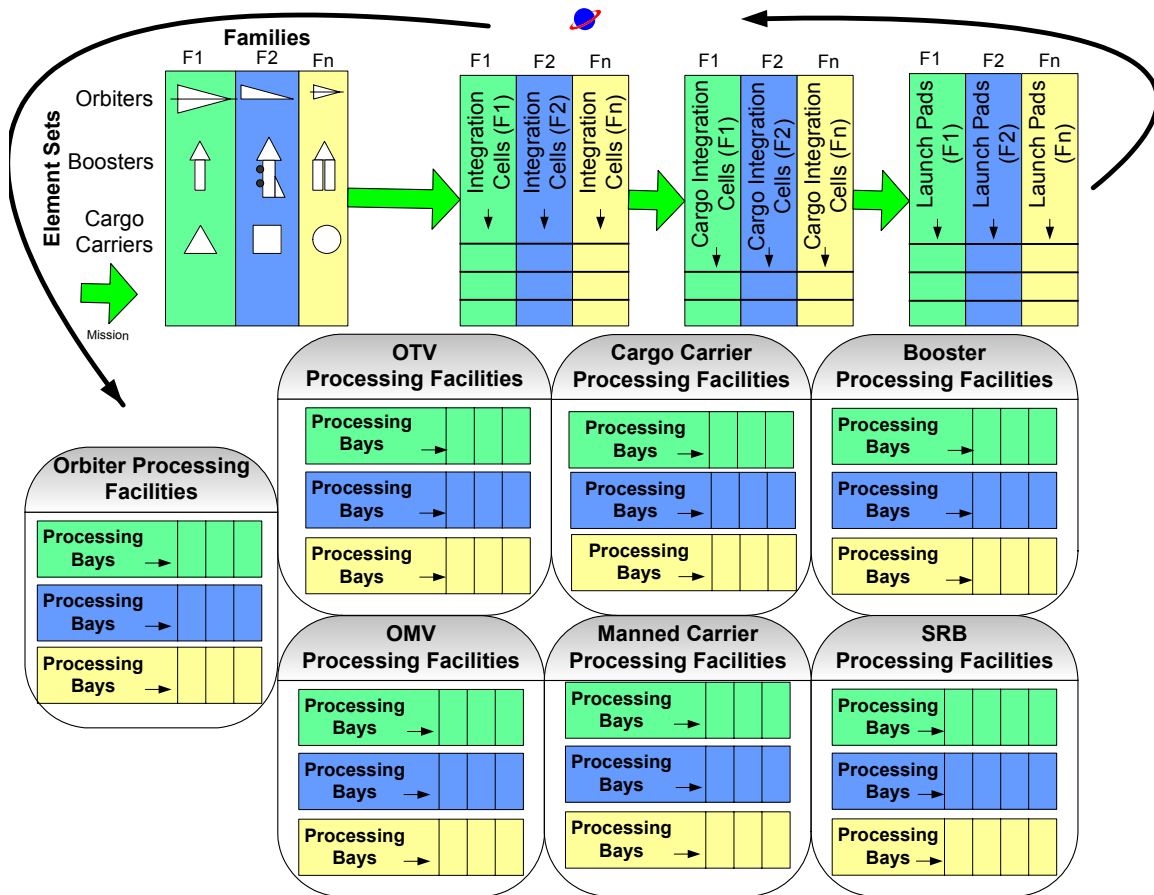
### The Architectural Model

A discrete-event simulation model was implemented to achieve the goals of this project. Several reasons justified the use of discrete-event simulation to model the problem at hand including its ability to model complex and dynamic discrete systems, and taking uncertainty into account. ARENA was used to implement the model.

Figure 4 corresponds to the same flow shown in Figure 3 but with more details that capture the Launch System Structure defined earlier. When a mission is generated and released to the system, it comes with attributes enforced by the manifest (launch date, payload weight, and destination). Also, the Family and Configuration numbers are attached to the mission as attributes. Based on these attributes (i.e., Family and Configuration), the mission seizes the appropriate elements with the appropriate amounts. For example, if the mission must be accomplished with a Configuration from Family 1 (F1 in Figure 4), then it will choose Elements from that family only based on the Configuration requirements. So for example, if the Configuration consists of an Orbiter and a Booster, it will seize two of these elements that belong to F1.

As missions progress forward, they go through Integration, Cargo Integration, and Launch Operation seizing the appropriate resources (i.e., facilities) based on Family number. Note that in these three stages delay times are based on Configuration-dependent statistical distribution to reflect uncertainty existing in the real system. As a

result, the launch vehicle may not always be ready for service at the launch date. The consequence of not meeting the launch date will be assessed via a delay cost that will be discussed later. In the same token, a mission may arrive to the launch pad earlier than its launch date. No mission is allowed to launch before its launch date, and therefore, the mission will be kept on the launch pad until its launch date. In the meanwhile cost will be incurred since launch resources will remain tied up with the mission until launch.



**Figure 4.** Process Flow in the Model

After launch, expendable elements are disposed and reusable elements are suspended from use in the model until the mission time in orbit has elapsed. Depending on the manifest, the mission may be sent to LEO, ISS, GEO, Planetary, or Polar. Reusable elements are continuously tracked on Orbit until the mission is completed, and they are routed to Landing. Note that the model can have more than one mission on Orbit at the same time. This can be easily prevented by making a simple change to one of the model inputs if desired as will be explained later.

After Landing, the Elements are Safed (in a Safing facility) and Processed to get them ready to be used with the new missions. It is important to note that at the Landing, Safing, and Ground Processing stages, the delay times are per Element and *not* per

Configuration. The reason for doing this is that the Elements may land separately and they actually get processed separately. For example, if a Configuration consists of an Orbiter and a Booster, the Booster may land a few minutes after launch, while the Orbiter may spend several days on Orbit before landing. Obviously, the Booster's Ground Processing will most likely start earlier than the Orbiter. Also, the type of Safing and Ground Processing facilities is typically Element-dependent. Therefore, the Orbiter would be sent to the Orbiter Processing Facility, while the Booster would be sent to the Booster Processing Facilities. Within each processing facility type, there are multiple processing bays that can be entered by the user.

As refurbishment completes in the Ground Processing area, the elements are freed and made available for future mission to seize them again.

### Model Components

The process described above requires the following components to be implemented.

#### *Launch Elements*

A generic set of Expendable and Reusable Elements shown in Table 1 is predefined in the model, and if any new Element types need to be considered, the model must be modified:

**Table 1.** Launch Elements

<b>Reusable Elements</b>	<b>Expendable Elements</b>
Orbiter	Expendable Orbiter
Booster	Expendable Booster
OMV (Orbital Maneuvering Vehicle)	Expendable OMV
OTV (Orbital Transfer Vehicle)	Expendable OTV
Cargo Carrier	Expendable Cargo Carrier
Manned Carrier	External Tank
SRB (Solid Rocket Booster)	

Note that these elements are generic enough to handle all future vehicle concepts that NASA is considering. Also, in this model we are interested in the functionality of an Element rather than its design detail. For example, an Orbiter is a reusable element that can carry cargo and humans to space, accomplish the mission and come back. As long as we know its time and cost information, the model does not its other specifications such as size, material, manufacturer, etc.

#### *Model Inputs*

Inputs to the Architectural Model include the Manifest, Family definition, Configuration definition, processing times, and cost factors. These inputs are discussed in greater detail here.

### Manifest

The manifest may include more information than what the model needs for other purposes. However, what is discussed here are only those inputs required by the simulation model. A manifest sample is shown in Table 2.

**Table 2.** Manifest Sample

Mission	PL Weight	Destination	Delay Penalty per				
			Release Day	Launch Day	Day per mission	Family	Config
1	30000	1	0	30	0	1	1
2	50000	2	20	60	0	1	1
3	30000	1	50	130	0	1	1
4	30000	1	120	180	0	1	1
5	30000	1	170	235	0	1	1
6	50000	2	225	290	0	1	1
7	30000	1	280	340	0	1	1
8	50000	2	330	395	0	1	1
9	50000	2	385	445	0	1	1

- **Mission No.** is a serial number for each mission to be able to track a specific mission if necessary.
- **P/L Weight:** is the payload weight. This input is needed to assign the appropriate Configuration to a certain mission. This assignment is done by the user at this point, and therefore, this input is not directly used by the simulation; however, it is kept in the model for future use if necessary.
- **Destination:** is where the mission is headed. This mission attribute is also needed to assign appropriate Configuration to a certain mission and must be entered by the user. It is include in the manifest for future use if necessary. Note that ARENA internally translates string attributes to numbers. However, when these attributes are coming from an external source such as a spreadsheet, they must already be in numeric format. Therefore, the destinations were encoded beforehand as shown in Table 3.

**Table 3.** Destination Code

Destination	Code
ISS	1
LEO	2
GEO	3
Planetary	4
Polar	5

- **Release Day:** is the date when the mission is released to the system to start the flow. This date can be arbitrarily entered by the user, or can be calculated by estimating the time needed in each facility and subtracting that from the Launch Day. This is rather a simple heuristic to generate mission release times. More sophisticated methods can be applied in the future such as a date that is dependent on the real-time dynamics of the system.
- **Launch Day:** is the planned launch day for a specific mission. A mission can *not* be launched prior to this date. The model allows for launch delays; however, there is a penalty associated with such delays.
- **Delay Penalty per Mission:** is a daily penalty for missing the launch date. One can think of this penalty as tardiness penalty. Delay cost comes from delay/mission and delay/Configuration. The default values for delay/mission have been currently set to zero, but the field is kept for future use.
- **Mission Support Cost per Mission:** is a daily cost for mission support. Mission support cost starts adding one day prior to launch until the end of mission. In this model, the mission support cost factor is a linear combination of mission support cost/mission and mission support cost/Configuration. The mission support cost/mission has been set to zero, but the field is kept for future use.
- **Family:** is the Family number to which the vehicle flying the mission belongs.
- **Configuration:** is the vehicle number that will fly the mission. Note that Configuration numbers are unique regardless which Family they come from. For example, if there are two families and each one has two configurations, then Configurations 1 and 2 will belong to Family 1 and Configurations 3 and 4 will belong to Family 2.

### Processing Times

Processing times are the times spent by vehicles in different facilities. Processing times have two levels of fidelity. In the Integration, Cargo Integration, and Launch Operations areas, the processing times are per Configuration. However, when reusable elements go to Orbit and then land, they go to different facility types, and therefore, the processing times are per element. Both Configurations and Elements seize the necessary facilities based on the families they belong to.

No historical data is available for any of the future vehicles and therefore deterministic or stochastic delay times are based on expert judgment. The only historical data available is the Shuttle's data, which will be used as a case study.

### Cost Information

The model consists of a generic set of processes that uses top-level information from studies of individual launch systems to assess the costs of using an architecture to meet the manifest requirements. The model includes the following cost components:

- 1) **Element Cost per Use:** is the “cost per use” of an element. For an expendable element, it is the procurement cost for that element. The “cost per use” for a reusable element is derived by dividing its design life by the number of flights an element is expected to fly.
- 2) **Processing cost:** is a rate per configuration per facility multiplied by the processing time at that facility. This cost is generated in the Integration, Cargo Integration, Launch Operations, and Ground Processing areas. As the mission goes through the first three facilities (Integration, Cargo Integration, and Launch Operations), the cost factors are per configuration; however, when the reusable Elements land back, the Processing cost at the Ground Processing Area will be the cost factor per Element multiplied by the processing time an Element spends at that processing facility. It is very important not to confuse the term “Processing Cost” with the “Ground Processing Cost”, which is only one of the four processing areas.
- 3) **Delay (tardiness) cost:** is the daily cost for missing the launch date. This cost is broken down to cost factor per mission multiplied by the number of tardy days, and cost factor per configuration multiplied by the number of tardy days. This cost is accumulated for all missions in a manifest and can be represented by:

$$DC_i = \alpha_i \max(A_i - P_i, 0) + \beta_c \max(A_i - P_i, 0)$$

where

$DC_i$  : Delay cost for mission  $i$

$A_i$  : Actual launch day for mission  $i$

$P_i$  : Planned launch day for mission  $i$

$\alpha_i$  : Delay penalty per day per mission  $i$

$\beta_c$  : Delay penalty per day per selected configuration  $c$

The total delay cost would then be:

$$DC = \sum_{i=1}^n DC_i$$

where

$n$  : Number of missions in a manifest

The parameters  $P_i$ ,  $\alpha_i$ , and  $n$  are obtained from the manifest (see Table 2). The actual launch day for a mission  $A_i$  is decided by the simulation. As a result, the value of the Delay Cost DC will be generated by the simulation.

- 4) **Mission Support Cost:** is Mission Support Cost factor per Configuration multiplied by the time a day prior to launch until landing.



### Simulation Model Implementation and Results

The discrete event simulation was implemented in ARENA, a simulation modeling environment from Rockwell Software. The manifest inputs and cost factors are saved in MS-Excel spreadsheets for easier modification of data.

The model results that were verified by experts in the field. The only available historical data during this phase was the Shuttle's data, which was populated in the model and produced reasonable results. More experiments are necessary to ensure model validity.

In Phase II of this project, the Architectural Model was extended to study the initial feasibility of utilizing the existing expendable fleet of Atlas and Delta launch families to accomplish the lunar mission as initiated by the by the president of the U.S.A. in 2004.

## **PHASE II: RESEARCH EXTENSION OF THE ARCHITECTURAL SIMULATION MODEL FOR EXISTING AND FUTURE LAUNCH SYSTEMS\***

### **Abstract**

The Architectural Simulation Model developed in Phase I was extended in Phase II to facilitate studies of manifest modularity options. The modified model is capable of simulating both reusable and expendable launch vehicles, and is able to distinguish between various launch vehicle models and families in terms of cost, readiness requirements, and reliability. The primary outputs of the model include lunar mission launch costs and mission success metrics. The manifest modularity studies that the model was modified to support focus on crew and cargo delivery options for recurring lunar missions in the 2015-2020 time frame.

The modified model contains logic to simulate the integration, pad, ascent, on-orbit, descent/landing, and processing phases of typical launches, and has the ability to differentiate between paths through these phases that reusable and expendable vehicles require. The model also has the ability to track payloads from multiple launches as they are integrated together and launched as single lunar missions from Earth orbit to trans-lunar injection orbits.

Three launch-specific decision points determine if payloads are successfully delivered and integrated in the proper orbits, and in the case of failures the model schedules re-flights appropriately. These three decision points model vehicle launch failures, payload delivery failures, and payload integration (rendezvous and docking) failures.

The primary outputs of the model include overall lunar mission launch costs, initial lunar launch target success, and overall mission success. The overall costs include the launch costs associated with all flights supporting a particular mission including re-flights, with two distinct costing techniques utilized to distinguish between commercial and government launches. The initial lunar launch target success pertains to the probability that an overall mission was integrated on-orbit and launched by a specific launch target date. The overall mission success pertains to the probability that an overall mission was integrated and launched at any lunar launch window opportunity before the effects of orbit degradation become significant.

Probabilistic distributions were applied to the various phases in order to model integration, processing, and pad time uncertainty. The three decision points were assigned probabilities based on launch vehicle characteristics as well as forecasted future capabilities. The model will be replicated in a stochastic manner to obtain meaningful output values with associated confidence intervals.

---

\* The modifications to the Architectural Model in this phase were mainly implemented by John D. Reeves Jr. during his LARSS program at LaRC.

The model's output metrics will be used to study differing levels of component modularity amongst lunar mission manifests. The various levels of modularity should lead to smaller launch count manifests, which should translate to increased mission reliability as well as lower lunar mission launch costs.

### **Introduction**

The focus of this phase of the project was to analyze the effects of varying the level of modularity used in a lunar mission manifest in order to see the resulting impact on cost and success metrics. The *Architectural Model* that was developed in Phase I of this project was extended to conduct this study. The Architectural Model simulated generic launch preparation operations for both expendable and reusable access-to-space vehicles, with cost and tardiness being the main output statistics. The model accepted mission manifests as inputs, dictating launch dates as well as what launch vehicles to use. The model simulated everything from the initial seizing of launch vehicle components (SRBs, orbiters, expendable boosters, etc.) to the orbit arrival and return to earth phases of each of the components. As NASA focus began to adjust to the Bush initiative of returning to the moon, it became apparent that various studies would be needed that provided insight into the optimal ways of getting cargo and crew to the lunar surface. Since a popular lunar mission scenario consisted of various mission pieces, or components, being integrated on-orbit prior to a trans-lunar injection burn, the levels of modularity, or component breakdown, was identified as a primary area to investigate. Various levels of modularity generally lead to different launch requirements, which in turn have a direct impact on the overall mission's cost and success probability. A modified version of the Architectural Model was identified as an appropriate tool to use in order to analyze these effects of varying lunar mission modularity. The Architectural Model only simulated various vehicles reaching orbit and returning after a designated on-orbit time, so the model had to be expanded in order to encapsulate various lunar mission components, in the form of payloads, rendezvousing and integrating with each other before a lunar mission on-orbit launch. The model also needed to be updated with various real-world circumstances, such as loss of vehicles (LOV) scenarios, payload delivery failures, and on-orbit rendezvous and docking failures. Supporting work included research focusing on current launch vehicle capabilities and costs to be used to update the model with the proper vehicle fleet.

### **Architectural Model Extension and Modification**

The initial purpose of the original Architectural Model was to simulate all ground operation phases of a reusable or expendable access-to-space vehicle. An Excel-based manifest was used to designate what vehicles are launched on what launch dates, and the model stochastically simulated these events and returned tardiness and launch cost metrics. The primary phases that were modeled include component integration, launch pad operations, ascent, on-orbit operations, landing and safing, and component processing. The ascent and orbit phases were modeled only to the detail of time durations and component separations and returns. The model was primarily used to analyze life-cycle cost metrics for governmental access-to-space vehicles such as the Space Shuttle, so the launch costing mechanism used the time durations in each of the facilities/phases in conjunction with designated "per day" factors in order to determine

the individual launch costs. This model appropriately modeled all of the main turnaround and cost drivers associated with reusable launch vehicles, and provided a helpful way to analyze various manifests. Description of the model to a greater detail was discussed in Phase I of this report.

The modularity study required that the Architectural Model be modified in a number of ways, some being significant. Some of the terminology used within the model also changed to reflect the focus transitioning from individual launches to an overall lunar mission. For example, the term mission specifies a lunar mission, which may consist of numerous launches. Each launch from the Earth's surface may have multiple payloads, which integrate with other payloads from other launches prior to the mission's launch from Earth orbit to the moon via a trans-lunar injection burn. A manifest is the Excel-based schedule of launches that support a single lunar mission. A manifest can technically contain multiple lunar missions and their associated launches, which was not required during this study. The following notes document some of the major changes that were made to the model, along with the reasoning and need:

- The costing mechanism was updated to account for fixed launch costs in addition to factor-based costs. Commercial launch companies such as Boeing and Lockheed Martin typically charge a fixed fee to place a payload in orbit, regardless of how long launch vehicles spend in the various readying facilities. If the launch is a governmental launch, such as the Space Shuttle, the launch costs are typically based on how long vehicles spend being integrated and processed. This update allows the model to use either of these two methods, or a combination of the two, to capture any sort of launch cost.
- The ascent and on-orbit phases were remodeled/modified to capture the delivery of payloads to orbit. Previously, the model only captured the ascent and on-orbit time durations of various vehicle components without any payload visibility. These modifications allow the model to simulate the orbit arrival of single or multiple payloads, as well as the rendezvous and docking of each. A recurring window in the form of a resource was also added that represents the opportunity of performing trans-lunar injection burns to send integrated vehicles to the moon. These modifications also allow the model to track how long individual payloads have been on orbit, which is later compared to a maximum on-orbit duration.
- The model was updated to track two different success metrics associated with integrating a mission's payloads while on-orbit prior to a trans-lunar injection burn. The first of the two success metrics, the *Target Success*, pertains to the probability of meeting a specified trans-lunar injection burn target. This target is estimated in conjunction with the manifest development, and takes into account a delay buffer for each of the launches that make up that particular mission. To clarify, a target date is chosen in such a way to allow up to a 20% delay in any or all of a mission's launches and still meet the target lunar launch date. It should be noted that the launch target corresponds to an opening in the recurring lunar launch window, which for this study has been assumed to be every nine days due

to a particular Earth/Lunar alignment scenario. The second of the two success metrics, the *Mission Success*, pertains to the probability of successfully integrating and launching a mission at any of the recurring lunar launch windows before an orbit degradation window is surpassed. Orbit degradation becomes a constraint since payloads placed in orbit will experience orbit decay over time that may prohibit successful integration. An orbit-degradation window of 150 days has been assumed in this study, which is a conservative estimate considering the manifests are calling for launch deliveries to 220nm orbits.

- Three major decision points were added to reflect failures associated with delivering payloads to orbit. The first of these failures is a loss of vehicle (LOV) decision that reflects the probability of a vehicle experiencing a catastrophic failure during launch or ascent. This decision point uses LOV reliabilities specified for each of the vehicles in the model. Logic was also added that schedules a replacement launch any time a launch is lost. A second decision point pertains to payload delivery failures. This reflects scenarios where a vehicle has a failure that does not lead to a loss of vehicle, but does lead to a failure of delivering a payload or multiple payloads to a targeted orbit. In this case the model reschedules another launch with the payloads that were not successfully delivered. The third decision point in the model pertains to the failure of an individual payload to integrate on-orbit. In this scenario, another launch is scheduled with the lost payload.
- The model was also modified to write to manifest Excel sheets the resulting launch times for single model replications. This allows users to see which launches were delayed and which were the primary drivers in the overall delivery and integration time.

### **Modularity Study**

The term modularity is used to denote varying levels of component breakdown for lunar mission scenarios. For example, a trans-lunar injection stage, or TLI stage, used to accelerate the integrated components to the moon can be broken down into several elements. Also, two elements such as an ascent stage and descent stage can be consolidated before launch to become a single payload element. Table 4 contains descriptions of the various elements that make up the lunar mission in the modularity study. It should also be noted that the four engine stages (TLI, TEI, DS, and AS) all have propellant associated with them. In some cases launch vehicle capability constraints dictate that a stage's propellants be launched on a separate vehicle, which leads to an extra payload that has to be integrated on orbit.

The modularity study called for a study of a varying level of modularity or component breakdown. It was decided that the modularity level would be swept from eleven elements to four elements. The reason for this was that a smaller level of modularity was thought to result in a lower cost and higher success rate since fewer elements would have to be integrated on orbit. Table 5 contains the element breakdowns for each of the different levels of modularity of interest. The elements that are combined with a “/”,

such as “EV/TEI” pertain to elements that are physically integrated on the ground prior to launch and do not require an on-orbit rendezvous and docking.

**Table 4.** Lunar Mission Component Descriptions

Acronym	Name	Description
EV	Entry Vehicle	Vehicle used by crew to re-enter Earth atmosphere
SH	Surface Habitat	Lunar surface habitat for crew
OM	Orbiting Module	Modules used during trans-lunar phase of mission
TLI	Trans-Lunar Injection Stage	Engine stage used to accelerate integrated elements to moon from Earth orbit
TEI	Trans-Earth Injection Stage	Engine stage used to accelerate crew back to Earth from lunar orbit
DS	Descent Stage	Engine stage used during lunar descent
AS	Ascent Stage	Engine stage used during lunar ascent

**Table 5.** Modularity Level Element Breakdowns.

Number of Elements	Element Breakdown
11	EV, OM, SH, AS, DS, TEI, 5 TLI
10	EV, OM, SH, AS, DS, TEI, 4 TLI
9	EV, OM, SH, AS, DS, TEI, 3 TLI
8	EV, OM, SH, AS, DS, TEI, 2 TLI
7	EV/TEI, OM, SH, AS, DS, 2 TLI
6	EV/TEI, OM/AS, SH, DS, 2 TLI
5	EV/TEI, OM/AS, SH/DS, 2 TLI
4	EV/TEI, OM/AS, SH/DS, 1 TLI

In addition to the eight different levels of modularity that were analyzed, six different launch vehicle fleet scenarios were considered. The first scenario focused only on the vehicles that Boeing will be operating in the 2015 time frame, such as the Delta IV HLV. The second scenario focused on only the vehicles that Lockheed Martin will be operating in the 2015 time frame, namely the Atlas V HLV. The third scenario was a combination of the first two, which basically encapsulates all of the vehicles that will be operating around that time frame that have been defined today. The fourth, fifth, and sixth scenarios focused on this same fleet of Boeing and Lockheed Martin vehicles in addition to advanced concepts. The advanced concepts that were used were a shuttle derived concept (82 mt to a 220nm) and two expendable launch vehicle designs (51mt and 105mt respectively to a 220nm orbit). These of these three advanced concepts has estimated cost and turnaround times that were incorporated into the model.

The eight levels of modularity and six fleet scenarios lead to a total of forty-eight combinations that were run through the modified Architectural Model. For every combination, a manifest had to be developed that specified what payload elements were launched on what vehicle along with the associated launch dates. The manifesting was somewhat difficult in that the number of launches associated with each of the combinations did not usually reflect that combination's level of modularity. Because of the payload delivery capability constraints of the launch vehicles, propellant sometimes had to be launched separately from corresponding dry pieces in order to logically schedule the flights. This sometimes led to a different launch count trend than anticipated. Usually a decrease in modularity would lead to a decrease in the launch count, but sometimes the redistribution of elements due to the modularity change would cause an increase, especially for fleet scenarios that focused on launch vehicles will relatively smaller payload capacities.

### **Conclusions**

All of the manifest combinations were successfully run using 500 replications, and some significant trends were apparent in the resulting output metrics. One finding was that the manifesting and architecture selection usually had a greater impact on the resulting cost and success metrics than the level of modularity. The allocation decisions made while creating the manifest usually dictated the cost and success metrics more so than the modularity level. The primary driver for this was that the number of elements that had to be integrated on orbit did not vary monotonically as the modularity was swept since the mission elements often had to be separated into wet and dry segments. Out of the six fleet scenarios that were considered, the current fleet and the Shuttle-derived concept fleet stood out in terms of the cost and success metrics. This finding is significant because it demonstrates, based on the various assumptions made in the model, that the current fleet of expendable launch vehicles is capable of supporting the new moon initiative. It is also important in that it identifies the Shuttle-derivative concept as the superior of the advanced concepts that were studied.

The modified Architectural Model proved to be an adequate tool to utilize in this study since it easily generated estimates of the cost and success metrics based on the manifest inputs. This study also demonstrated the benefits of using discrete event simulation in general and serves as an example of the growing need of DES tools in the access-to-space industry. Future work consists of further evolving the model in a variety of ways in order to facilitate further studies in the lunar mission area or broader studies concerning any sort of access-to-space mission requiring multiple launches.

### **Future Work**

Among the most natural extensions of this model are:

- It will be interesting to have more experts inspecting the results of the model and approve its results. The model will gain more credibility if its results are compared against other tools that NASA uses to predict the cost of future architectures and their ability to meet the schedule.

- Other types of costs can be included in the model such as *earliness* cost, which is a cost for each day a mission is ready to launch but can not because it is early to its launch date.
- The missions can be generated based on more sophisticated methods and heuristics, which could be based on how these missions are generated in real life.
- Simulation Optimization: The current model relies on having the user select vehicle concepts to accomplish the missions in a certain manifest. The model will produce two main results. First, it will show if the manifest schedule can be met using certain architecture, and if it is not, by how much it was delayed. Second, it will provide the cost of each mission, as well as the cost of accomplishing all missions in the manifest. An interesting extension of this model is to overlay an Optimization Model that is capable of defining the mix of launch systems that best supports a mission manifest when no pre-defined architectural mix of launch systems exists. This can be done by matching the missions in a manifest with the concepts that would meet the manifest schedule at the minimum cost (given that multiple systems could perform the same mission). The Optimization Model, however, must have control over concept selection, resource acquisition, and facilities capacities. The outcome of the model would be the optimal (or near optimal) cost and schedule. The benefit of this extension would be the ability to compare the optimal total cost of using a specific architecture to support a specific mission manifest with the optimal total cost of using alternate architectures.

In Phase II, an Experimental Design approach was conducted to select the optimum levels of inputs. It will be useful to apply similar methodology to the generic Architectural Model developed in Phase I.

- Evolving the model in a variety of ways in order to facilitate further studies in the lunar mission area or broader studies concerning any sort of access-to-space mission requiring multiple launches